

Hallo Welt!

GmbH

## Fundamental Security

Hallo Welt!

GmbH

„Learn more about SSL, Authentication, Extensions, API, External Data,  
Common.js“

Hallo Welt!

GmbH

## What is this all about?

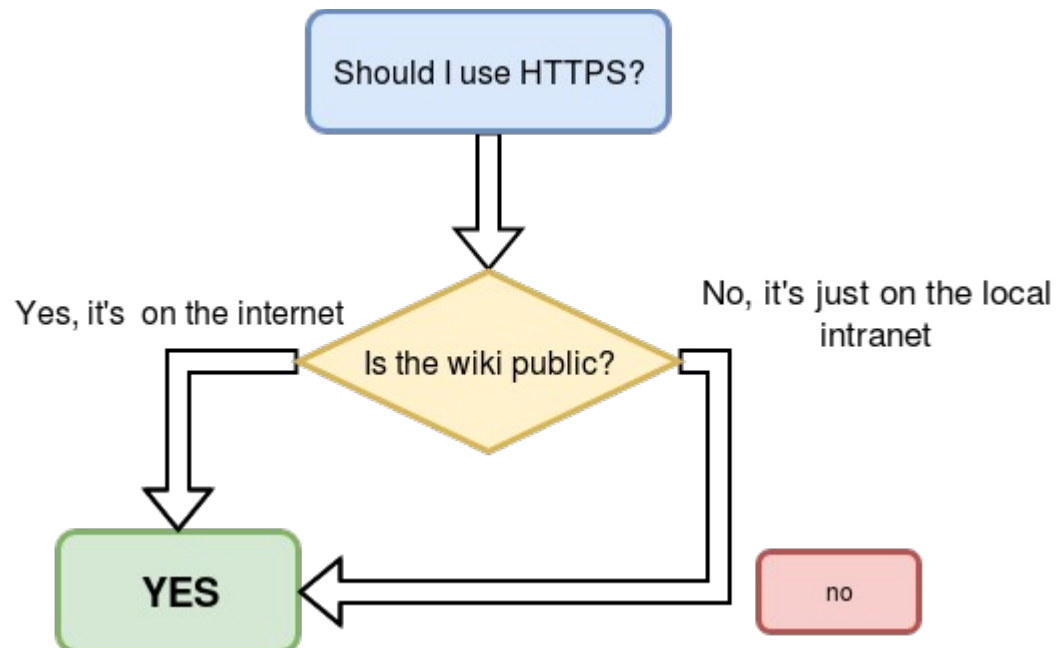
- Server / Network security? → A little bit
- Application / Code security? → Yes
- Content security? → Yes

Hallo Welt!

GmbH

**Server / Network security → SSL**

## Server / Network security → SSL



Hallo Welt!

GmbH

## Server / Network security → SSL

- Wiki on the internet → At least use **Let's encrypt** (<https://letsencrypt.org>)
- Wiki on the (business) intranet → **Selfsigned** or better from a CA



## Server / Network security → Server access

- Disable root login
- Use key-based authentication
- Switch SSL port to „non-standard“
- Use fail2ban



## Server / Network security → OS and services

- Make sure „services“ do not allow outbound connections. Esp. **ElasticSearch** and MySQL
- Run unattended updates (if possible)
- Do proper monitoring (e.g. „icinga“)



Hallo Welt!

GmbH

## Server / Network security → Application access

- Restrict access to the wiki application using
  - Kerberos (business intranets) or
  - client SSL certificates (cloud)
- Be aware: Services might need to bypass



## Application security

- Use long term support (LTS) versions
- Use WMF extensions (proper security review!)
- Use actively maintained extensions only

Hallo Welt!

GmbH

## Application security

- Make sure „MediaWiki:Common.js“ can only be edited by trustworthy people
- Same for „Gadgets“ and „Lua“-Modules
- When using „External Data“ be aware of the „proxy user priviledges“



## Application security

- Do simple security audits yourself
  - After activation, check the network panel in the browser's developer tools
  - Check for direct database access (`"->select(", "->query(")`)
  - Check for object model content access (`"->getContent"`)
  - Check for proper permission checks (`"->userCan(", "->isAllowed(")`)
  - Check for proper escaping/sanitizing (`"Html::", "::sanitize"`)
- If writing extensions
  - implement proper permission checks
  - do proper input sanitizing
- → prevent information disclosure (e.g. in error messages)



## Content security → User authentication/authorization

- Use authentication/authorization extensions
  - SAML / Shibboleth
  - LDAP
  - OAuth / OAuth2
  - OpenID (Connect)



## Content security → Entry point configuration

- Secure uploaded files with `$wgUploadDirectory`, `$wgUploadPath` and `img_auth.php`
- Prevent access to all PHP files other than `img_auth.php`, `index.php`, `api.php`, `load.php`, `opensearch_desc.php`, `thumb.php` and `thumb_handler.php`
- Some extensions may load images, CSS and JavaScript files directly from `extensions/` directory
- Prevent access to (or remove) `.git/` directories
- Prevent access to configuration files in JSON format

## Content security → Access control

- Use access-control extensions
  - AccessControl
  - Lockdown
  - BlueSpice
- Be aware: some extensions may bypass such measures

Hallo Welt!

GmbH

## Resources

- <https://www.mediawiki.org/wiki/Manual:Security>
- [https://www.mediawiki.org/wiki/Security\\_for\\_developers](https://www.mediawiki.org/wiki/Security_for_developers)
- [https://www.mediawiki.org/wiki/Category:Page\\_specific\\_user\\_rights\\_extensions](https://www.mediawiki.org/wiki/Category:Page_specific_user_rights_extensions)
- [https://www.mediawiki.org/wiki/Manual:Preventing\\_access](https://www.mediawiki.org/wiki/Manual:Preventing_access)



Hallo Welt!

GmbH

**Thank you**